Name: _____ Date: _____

# Imitation, Prediction, Understanding

Reading level 14 · 2163 words

In 1950, Alan Turing opened a paper in the journal Mind with a question: "Can machines think?" Then he declined to answer it. If you settle the meaning of "machine" and "think" by how people use the words, he wrote, you would end up running a Gallup poll, which he called absurd. He proposed a game instead, the imitation game, because a game can be tested. In its simplest form, a judge types questions to two hidden players, one a person and one a machine, and tries to tell which is which. He judged the original question "too meaningless to deserve discussion," and guessed that by the end of the century, people would speak of machines thinking without expecting to be contradicted. This essay follows what became of his question, and of the machines built while it stayed open.

The name arrived five years later, in August 1955, when McCarthy, Minsky, Rochester and Shannon proposed a two-month study of "artificial intelligence" at Dartmouth College for the summer of 1956. Their conjecture was that every part of learning, or any other feature of intelligence, could be described so precisely that a machine could simulate it. Four years later, Arthur Samuel of IBM reported a checkers program that had learned to play better than the person who wrote it, starting from only the rules, a sense of direction and a list of features that might matter. Eight or ten hours of play were enough to improve it. In a narrow sense, it was learning from its own games.

How can a program be said to learn? In a neural network, layers of units pass numbers forward, and every connection carries a weight. The weights, plus a bias for each unit, are the parameters. Google's machine learning course counts 21 of them in a tiny network with three inputs, one hidden layer of four units and one output. Picture a large control panel with one dial for each parameter. Training starts with the dials set at random. It measures the gap between the network's output and the desired output, called the loss, and moves every parameter a small step in the direction that lowers it. Repeating that step is gradient descent. In 1986, Rumelhart, Hinton and Williams described back-propagation in Nature, a procedure for adjusting the weights so that hidden units "come to represent important features of the task domain."

Games became the proving ground because the score is clear, and, as one history of Samuel's work notes, it is easy to compare a computer's play with a person's. In May 1997, IBM's Deep Blue became the first computer to defeat a reigning world chess champion in a match under standard tournament controls. IBM says it won by brute force, with 32 processors judging 200 million positions a second. It gives the score as 3.5 to 2.5: two wins for Deep Blue, one for the champion and three draws. AlphaGo worked differently. Its 2016

Nature paper, which reports a 5-0 win over the European champion, describes networks that rate board positions and choose moves, trained on human expert games and then on games against itself, and joined to a search of possible moves. DeepMind counts a 4-1 win over Lee Sedol that March.

The next jump came from photographs. In the Toronto team's 2012 description, ImageNet held over 15 million labeled images in roughly 22,000 categories, collected from the web and labeled by people through Amazon's Mechanical Turk. The yearly contest used about 1,000 images in each of 1,000 categories. In 2012, Krizhevsky, Sutskever and Hinton trained a deep network with about 60 million parameters on 1.2 million of them, which took five to six days on two graphics processors. Averaging five such networks, the team's five best guesses missed about 16 times in 100 on the contest's test images. A version also trained on the full collection missed 15.3, against about 26 for the runner-up. The contest's own report calls this the major breakthrough, and by 2014 all the top contestants leaned heavily on such networks.

Language took longer, and a key step came in June 2017, when a team of eight authors posted "Attention Is All You Need." It proposed the Transformer, a network "based solely on attention mechanisms, dispensing with recurrence and convolutions entirely." Attention lets the network ask, for each token of its input, how much every other token matters to it. In Google's example, "The animal didn't cross the street because it was too tired," attention should weigh "animal" more than "street" for the word "it." Swap "tired" for "wide" and the weight should shift to the street. Many layers stack many attention heads, each starting from random values, so different heads can learn different relationships. Google's blog said the design needed less computation to train and suited modern hardware better.

A large language model is, at bottom, a predictor of tokens. A token is a word, a piece of a word or a single character. In English, one averages about four characters, so 400 of them make roughly 300 words. In the first phase of training, the model reads text and guesses the next token, with the real text as the answer key. Meta's 2024 paper on its Llama 3 models describes training this way. Google's course calls such models "essentially autocomplete mechanisms," like the autocomplete in email, only vastly larger. To write, a model rates every possible next token and then takes the likeliest. Or it draws at random from those above a threshold, which is why answers vary. In 2022, OpenAI's team wrote that bigger models are not inherently better at following a user's intent. After tuning with human examples and rankings, people preferred answers from a model with 1.3 billion parameters over those of the 175-billion original.

Why did scale matter? Here, scale means three things growing together: the size of the network, the amount of text, and the computing power spent on training. In January 2020, Kaplan and colleagues reported that a language model's loss falls as a power law with its size, its data and its training compute. GPT-3 followed that May, with 175 billion parameters and 300 billion training tokens. In March 2022, DeepMind found such models were

significantly undertrained. Its Chinchilla, with 70 billion parameters and four times more data, beat the 280-billion Gopher, and tokens should double whenever size does. In 2024, Meta published a 405-billion model trained on 15.6 trillion tokens. By our arithmetic, that is about 52 times GPT-3's tokens for 2.3 times its parameters. These are published figures from three groups, dated 2020 to 2024, and none of them is a current record.

Training and use are different jobs, and Google's course says training a large model takes "multiple months and enormous computational resources and electricity." Using one to answer also costs a lot, though larger training sets typically reduce what each answer costs. All that learning needs data from somewhere. OpenAI's speech model Whisper was trained on 680,000 hours of audio, which by our arithmetic is about 78 years of nonstop sound. To a computer, sound is numbers too. Apple's 2017 description of the "Hey Siri" detector says a phone's microphone turns your voice into 16,000 waveform samples a second, and a small network on the device turns those into a score for how likely it is that you said the phrase.

Predicting a likely token is not the same as checking a claim. NIST's July 2024 profile on generative AI calls the result confabulation, "confidently stated but erroneous or false content." In September 2025, researchers at OpenAI and Georgia Tech argued that part of the cause is how the machines are scored. Training and evaluation reward guessing over admitting uncertainty, like a multiple-choice exam on which a blank earns nothing and a lucky guess earns a point. Stanford's 2026 AI Index calls the unevenness a jagged frontier. AI can win a gold medal at the International Mathematical Olympiad, yet the top system reads analog clocks correctly just 50.1% of the time. That is one benchmark in one report, and it will move.

Models also reflect their examples, as a 2018 test of face tools shows. Joy Buolamwini and Timnit Gebru tested three company-made tools that guess gender from a face photo. Two common test sets were mostly lighter-skinned faces, and the makers had not said what examples their tools learned from. On a new, balanced set, the tools got up to 34.7% of darker-skinned women wrong, against at most 0.8% for lighter-skinned men. In 2019, NIST tested 189 face-recognition algorithms from 99 developers. Most showed differences between demographic groups, and false-positive rates often varied by factors of 10 to beyond 100. Both studies measured particular systems at particular times, and companies have changed theirs since. What lasts is the method of testing a system on the people it will be used on.

Here are a few plain facts, as of November 2026, for anyone who types to a chatbot. These tools can state wrong things with confidence, repeat unfair patterns from their examples, and show unsafe content. Common Sense Media, in guidance reviewed in June 2026, advises limited use beside a parent or caregiver for ages 6 to 12. It also says what you submit can become a permanent part of a system's data. It advises kids to avoid names, addresses or photos. In September 2025, the Federal Trade Commission asked seven companies how their chatbots, when acting as companions, use or share personal information from users' conversations. A researcher interviewed by UNICEF in July 2025 said AI companion apps

often collect photos, voice notes and location. Plain advice that goes past those pages: never type a full name, a home address, a school or a photo. Any answer that matters is a claim to check against a source you can open.

Now the argument that has outlived Turing's paper: can a system trained to predict text be said to understand anything? Melanie Mitchell and David Krakauer of the Santa Fe Institute describe a heated debate in the AI research community. They report a 2022 survey of 480 researchers in natural language processing. Asked whether a machine trained only on text, given enough data and computing power, could understand language "in some non-trivial sense," 51% agreed and 49% disagreed. That is almost exactly even. The survey ran in May and June 2022, so it records opinion from then, not now, and it says little about why each person answered as they did, which is where the real argument begins.

The skeptics make two arguments. Emily Bender and Alexander Koller wrote in 2020 that "a system trained only on form has a priori no way to learn meaning." Their octopus test imagines a deep-sea octopus tapping an undersea cable between two islands. It is expert at statistical patterns and learns to answer like one of the speakers. Then the other asks about a coconut catapult, and the octopus is stuck, because it has never seen what the words refer to. The second argument rests on experiments by other researchers, which Mitchell and Krakauer describe. A model called BERT scored near human level on a reasoning benchmark. But cue words such as "not" predicted the right answers, and when those cues were removed, its score fell to essentially random guessing. They call this shortcut learning.

The other side starts from performance. Mitchell and Krakauer write that those who would grant understanding to current models base their view on test results: on benchmarks built to assess language understanding and reasoning, large models often equal or beat people. Some also point inside the system. In March 2025, a research team at Anthropic that studies its own AI reported on a rhyming couplet. The AI appeared to settle on a rhyming word before it wrote the line, and when researchers removed the part of its inner state that stood for that word, it ended the line with another sensible rhyme. They call this planning ahead. They also note that their method captures only a fraction of what the model computes. It is one team's report on its own system. The Nobel Prize's guide to its 2024 physics award takes the cautious view: computers cannot think, though machines can mimic memory and learning.

Turing's date for his guess has passed. In 2022, about half of surveyed researchers still said that text alone could not give understanding in any non-trivial sense. Mitchell and Krakauer note that tests built to measure human understanding might not suit machines. They argue that progress needs a broader science of intelligence, one that can describe distinct modes of understanding along with their strengths and limits. Nobody can say yet what that science will find. Meanwhile, the loop underneath is plain. The system reads what came before, scores every possible next token, chooses one, and repeats. Each answer from a chatbot is that loop, run once for every token it writes, and the question Turing set aside is still open.

## Questions

**1. What did Turing do with the question "Can machines think?"**

A. He answered yes
B. He replaced it with a testable game
C. He said no one could ever ask it

**2. In training, what does gradient descent do with each parameter?**

A. Deletes it if it is wrong
B. Copies it into the next model
C. Moves it a small step in the direction that lowers the loss

**3. Why can the same chatbot give different answers to one question?**

A. It may take the likeliest token or draw at random among likely ones
B. It has several different brains
C. The question changes each time

**4. What does "significantly undertrained" suggest about the models DeepMind studied in 2022?**

A. They had too much training data
B. They were too small to start
C. They had too little data for their size

**5. What did the BERT example show about shortcut learning?**

A. Models always understand arguments
B. A model can score well by using cue words instead of understanding
C. Removing cue words made the score perfect

**6. In the 2022 survey, what share of researchers agreed that a text-only model could understand language in some non-trivial sense?**

A. About 51%
B. About 20%
C. About 90%

## Words to know

parameters: learned numbers
gradient: slope direction
token: text piece
confabulation: confident invention
benchmark: standard test
conjecture: working guess

# Answer key: Imitation, Prediction, Understanding

1. B. He replaced it with a testable game
2. C. Moves it a small step in the direction that lowers the loss
3. A. It may take the likeliest token or draw at random among likely ones
4. C. They had too little data for their size
5. B. A model can score well by using cue words instead of understanding
6. A. About 51%